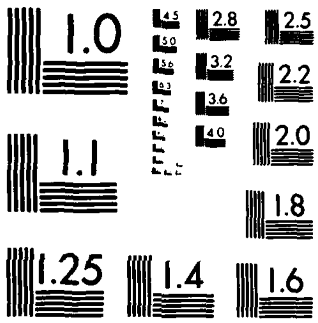
AD-A149 553 ORDER PICKING IN AN AISLE(U) GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION RESEARCH CENTER M GOETSCHALCKX ET AL. SEP 84 PDRC-83-11 1/1

UNCLASSIFIED N00014-83-K-0147 F/G 12/1 NL

END

2-85



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

AD-A149 553

Georgia Institute of Technology

ORDER PICKING IN AN AISLE

by

Marc Goetschalckx
H. Donald Ratliff

PDRC 83-11

PRODUCTION and DISTRIBUTION RESEARCH CENTER

SCHOOL OF INDUSTRIAL
AND SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM
OF GEORGIA
ATLANTA, GEORGIA 30332

84 12 05 002

PDRC Report Series 83-11
May 1984
Revised September 1984

ORDER PICKING IN AN AISLE

by

Marc Goetschalckx
H. Donald Ratliff

PDRC 83-11

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

This work was supported by the Office of Naval Research under Contract No. N00014-83-K-0147. Reproduction is permitted in whole or in part for any purpose of the U. S. Government.

Abstract

A classical order picking problem is the case where items have to be picked from both sides of an aisle and the picker cannot reach items on both sides simultaneously. Hence the picker must cross the aisle one or more times. Efficient optimal algorithms are developed for the cases where the picker enters and exits the aisle at the same end or at opposing ends. For all practical aisle widths and number of picks in an order, it is more efficient to enter and exit the aisle at opposing ends. The algorithms can be implemented in real time on a microcomputer. An optimal fixed picking sequence, suitable for implementation in a manual system is also developed and compared with the optimal policy.

1. INTRODUCTION

One of the most frequently encountered order picking problems occurs when items of an order have to be picked from both sides of an aisle and the picker cannot reach items on both sides simultaneously. To fill a customer order, the picker must cross the aisle one or more times.

Since the warehouse generally consists of many such aisles there are two significant problems to be addressed. One problem is to determine how to optimally travel from one aisle to another. For the case of parallel narrow aisles with crossovers only at the ends of the aisles (Figure 1), this problem was efficiently solved in Ratliff and Rosenthal (1983). They only considered the narrow aisle case where a picker can reach both sides simultaneously. However, their algorithm is optimum for any parallel aisle system provided that the optimum picking policies within each aisle can be determined. The second significant problem is to determine optimum picking policies within wide aisles. This is the problem addressed here. In addition a heuristic for the case of wide aisles suitable for manual implementation is presented and compared with the optimum procedure.

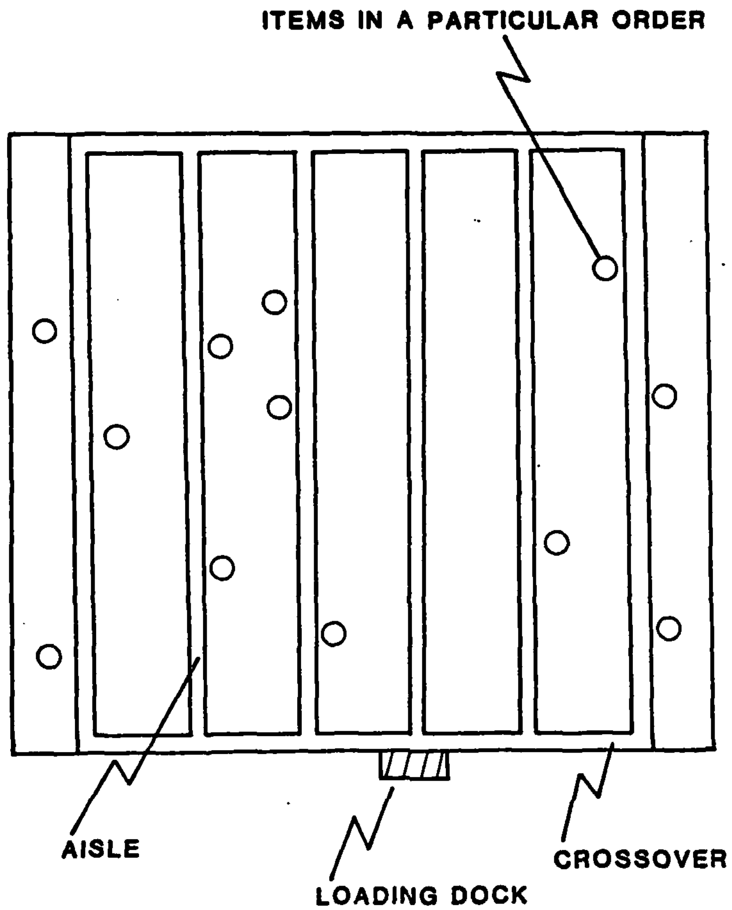


Figure 1. Warehouse Aisle Configuration.

For the results presented here, it will be assumed that the picker moves only on the aisle floor and travel from one position to another within the aisle occurs along a straight line connecting the positions. It will also be assumed that aisles are entered and exited at points half way between the sides of the aisle. The objective is to find the picking sequence with the shortest travel distance for picking the order.

The solution for the general problem under the above asumptions is the shortest Euclidean distance Hamiltonian path or circuit (i.e., the traveling salesman problem). It is well known that this is a difficult

problem to solve optimally, Karp (1982) and Garey et al. (1976). Fortunately, in the case of a single wide aisle, there exist additional structural properties, which allow an efficient algorithm. Other special cases of the traveling salesman problem which we can efficiently solve are given in Gilmore and Gomory (1964), Ratliff and Rosenthal (1983), Lawler (1971) and Cutler (1980).

2. DEFINITIONS

Two major classes of picking policies for an aisle are defined. A policy is called a "return policy" when the picker enters and leaves the aisle at the same end. A policy is called a "traversal policy" when the picker enters the aisle at one end and leaves it at the other end. If the aisle is part of a larger warehouse, then it might be efficient to pick part of the items from one end of the aisle and pick the remaining items from the other end of the aisle. This is called a "split return policy". Similarly, it might be efficient to pick part of the items while traversing the aisle in one direction and pick the remaining items while traversing the aisle in the other direction. This is called a "split traversal policy". The different types of policies are illustrated in Figure 2.

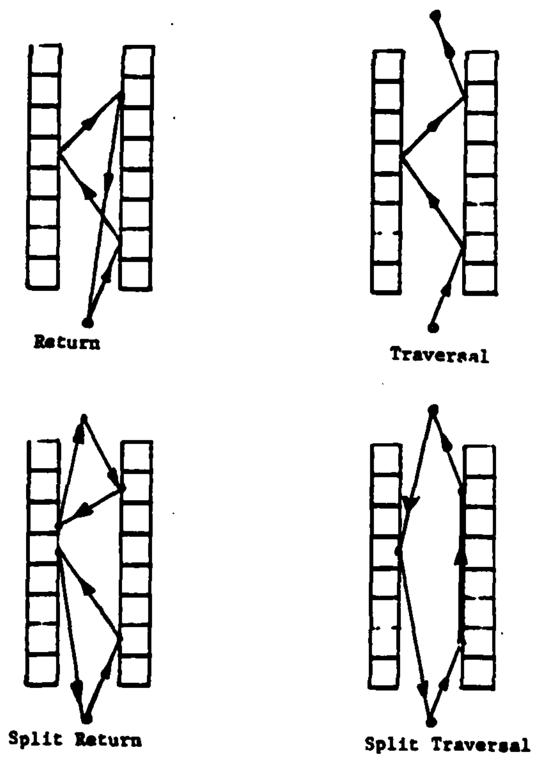


Figure 2. Return and Traversal Picking Policies

For this discussion we will assume that there are the same number of locations or slots on either side of the aisle and that all slots have the same dimensions. All of the properties and algorithms can be extended in a straightforward fashion to the case in which the slot dimensions are not the same or the number of locations on either side of the aisle is not the same.

The following definitions are illustrated in Figure 3. Let a be the width of one slot and let M be the number of slots on one side of the aisle. Let W be the width of the aisle measured in slot widths. The entry and exit points for the aisle will be located on the centerline of the aisle and 1/2 slot width outside the end of the aisle.

The number of items in one order will be denoted by N. Of those N items n will be assumed to be stored on the left side and m will be assumed to be stored on the right side of the aisle. On each side the storage locations are number from 1 through M starting from the near location. The location of the items will be given by the number of their storage loction. Items in the order on the left side of the aisle will be denoted by $L_1, L_2, \ldots L_n$. Items in the order on the right will be denoted by $R_1, R_2, \ldots R_m$. The distance along the axis of the aisle from the near entry point to item L_i and R_i is L_i and R_i slots respectively. For the remainder of this discussion we will use one slot width as the unit of measurement.

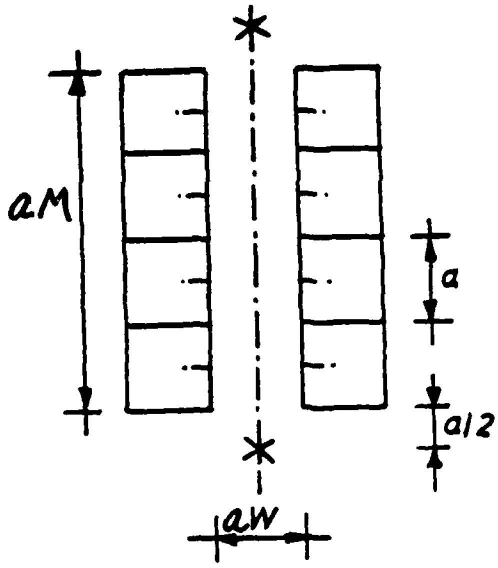


Figure 3. Aisle Dimensions

3. OPTIMAL TRAVERSAL PICKING POLICY

With a traversal policy, it is assumed that the picker starts the picking sequence at the near entry point and leaves the aisle at the far exit point. The following "no-skip" property allows the efficient determination of an optimal picking sequence under a traversal picking policy.

No-Skip Property

Before an item R_k can be picked in an optimal traversal picking sequence, all of the items $R_1, R_2, \ldots, R_{k-1}$ must already have been picked. The same relationship holds true between L_k and $L_1, L_2, \ldots, L_{k-1}$.

Proof. The proof is by contradiction. It relies on a well-known result first proved by Barachet (1957) which says that any shortest Euclidean distance Hamiltonian path or circuit does not cross itself.

Assume that there exists an optimal picking sequence, which has picked item R_k but has not yet picked item R_j, with $j < k$. The route started at the near entry point and is now at R_k, it still must visit the near point R_j and finish at the far exit point. There are four possible cases, depending on whether the previous point A in the circuit and the following point B are located at the same side of the aisle as R_k and R_j. These cases are illustrated in Figure 4. Each of these cases involves the route crossing itself (an overlap is a degenerate cross). Hence the picking sequence which ships R_j cannot be optimal. q.e.d.

A similar algorithm was developed by Cutler (1980) for planar traveling salesman problems, where all points lie on two or three parallel lines. Such problems occur in the design of printed circuits. In these problems the begin and endpoints also lie on the parallel lines, which makes these problems more complicated, compared to the order

picking problem discussed here. Cutler presented $O(N^2)$ and $O(N^3)$ algorithms for the two and three line problem, respectively. The algorithms are also based on the no-crossing property of Barachet.

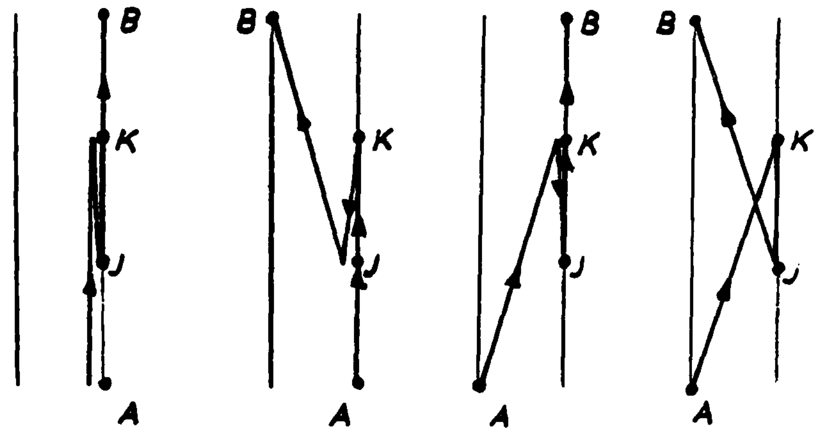


Figure 4. Four Possible Cases in Picking R_k before R_j.

The above property allows the characterization of the state of the system during an optimal traversal picking sequence by a 3-tuple (R_i, L_j, k), where R_i is the last item picked on the right side, L_j is the last item picked on the left side, and k is R or L, indicating whether the picker is currently on the right or left side.

For the entry point, R_i and L_j are equal to zero. For the exit point, either (R_i = n+1 and L_j = m) or (R_i = n and L_j = m+1), depending on whether k is equal to R or L, respectively.

For each state there exist only two state transitions if the picking is done by an optimal policy. Assume the system is in state (R_i, L_j, R). There are only two items that are candidates to be picked next. The

picker can either continue on the same side and pick the next item R_{i+1}, or he can cross the aisle and pick the next item L_{j+1}. The travel required for those transitions is given respectively by

$$t(R_{i+1},R_i) = R_{i+1}-R_i \tag{1}$$

$$t(L_{j+1},R_i) = \mathrm{SQRT}(W^2+(L_{j+1}-R_i)^2) \tag{2}$$

The travel for the transition from the entry node and the travel to the exit node for the left and right side respectively are given by

$$t(L_1,0) = \mathrm{SQRT}(W^2/4+L_1^2) \tag{3}$$

$$t(R_1,0) = \mathrm{SQRT}(W^2/4+R_1^2) \tag{4}$$

$$t(L_{n+1},L_n) = \mathrm{SQRT}(W^2/4+(M+1-L_n)^2) \tag{5}$$

$$t(R_{m+1},R_m) = \mathrm{SQRT}(W^2/4+(M+1-R_m)^2) \tag{6}$$

The problem of finding the optimal picking sequence is now reduced to finding the shortest path in an acyclic graph. The graph is illustrated in Figure 5. For each state (R_i,L_j,k) there is a node in the graph. For each feasible transition there is an arc in the graph between the corresponding nodes, with a length equal to the travel associated with the transition. The length of the shortest path is the total picking travel.

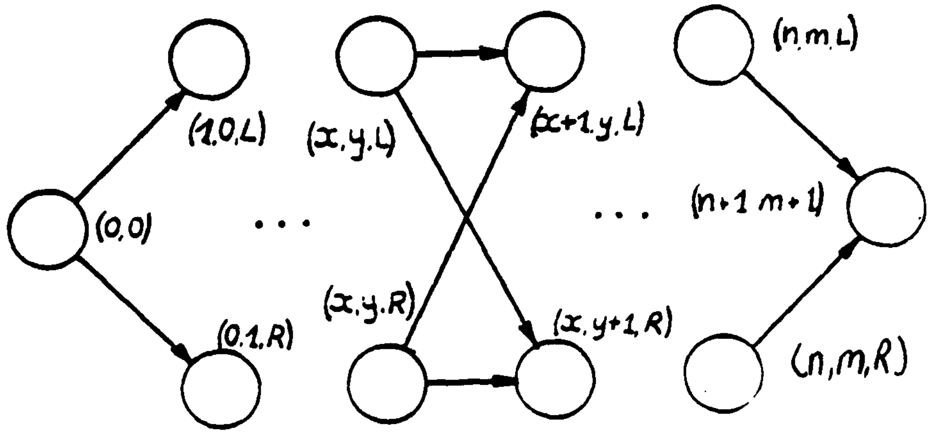


Figure 5. Shortest Path Graph for the Traversal Policy

There are a total of $(n+1)m+(m+1)n+2 = 2nm+m+n+2$ nodes in the graph. On the average, n and m will be equal to N/2. Any node has at most two outgoing and two incoming arcs. The computational effort to find the shortest path is then proportional to the number of nodes and thus the algorithm is $O(N^2)$ or quadratic in the number of items in an order. Several very efficient procedures exist for finding the shortest path in an acyclic graph, see Christofides (1975). It is necessary to sort the items by non-decreasing coordinates in order to specify the graph. This requires a computational effort of $O(n \cdot \log(n))+O(m \cdot \log(m))$.

An example is given by the two Figures 6 and 7, where $M = 10$, $W = 3$, $N = 5$, $n = 2$, and $m = 3$. The minimum travel to each state is given under its corresponding node, the travel length of each transition is given above its corresponding arc. The optimal picking distance is 19.39.

A pseudo code representation of the algorithm is given in Appendix A, together with a discussion of the comparison of computational effort required on different computer systems.

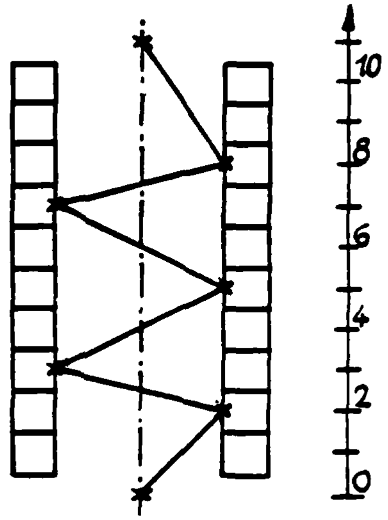


Figure 6. Example of a Traversal Sequence in an Aisle

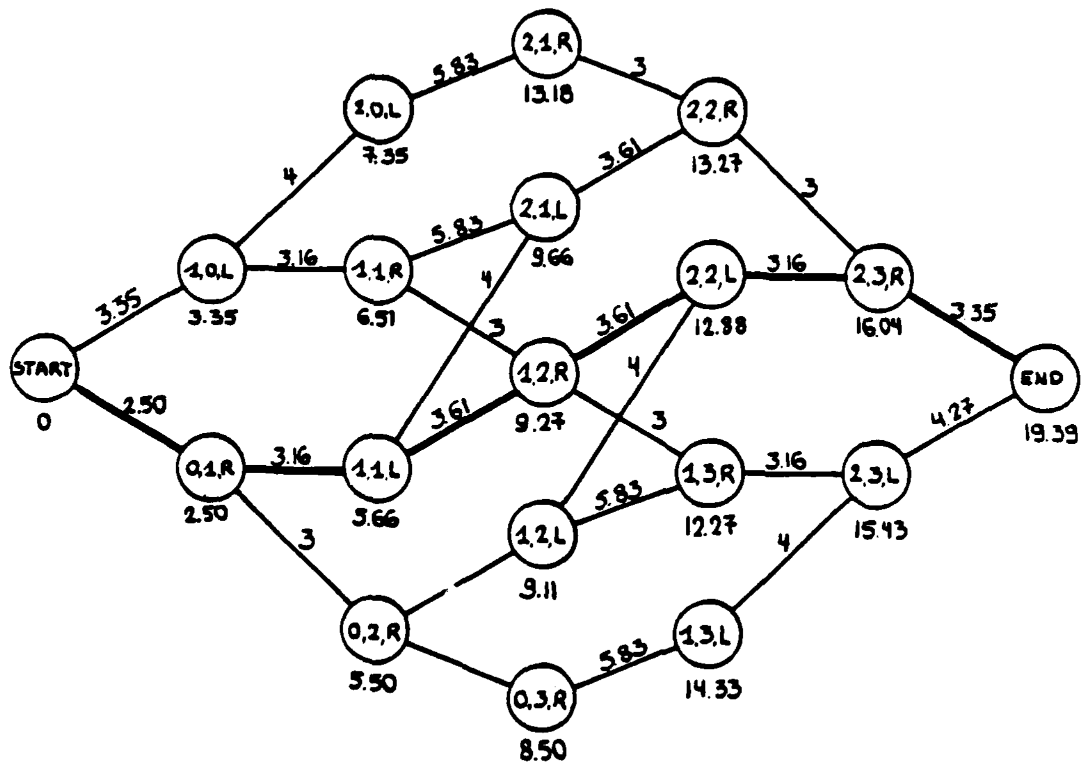


Figure 7. Shortest Path Graph Corresponding to Example.

The following experiment was conducted to examine the influence of the number of items in the order and of the width of the aisle on the picking time. The aisle had a length of 60 slots. Aisle widths of 2, 3, 4 and 5 slots were considered. The "order density" is the percentage of total slots in the aisle visited in picking the order. Order densities of 5%, 10%, 20%, 40% and 80% were considered. For each of these combinations, 10 replications were computed. The resulting average travel distance is given in Table 1. The width, the number of items and their interaction are significant factors in the travel distance at the 0.001 significance level.

Table 1. The Average Optimal Traversal Travel in Slots

	DENSITY					
	5%	10%	20%	40%	80%	AVG.
WIDTH						
2	62.6	64.7	71.8	86.3	109.7	79.0
3	63.8	69.8	79.8	98.5	123.3	87.0
4	66.0	72.5	84.6	107.3	134.7	93.0
5	67.3	75.1	92.9	119.1	146.1	100.1
AVG.	64.9	70.5	82.2	102.8	128.5	89.8

Any traversal picking sequence must at least travel from entry point to exit point, which is equal to the number of slots in the aisle plus one. We will call this the fixed travel. The variable part of the

travel is defined as the difference between the total travel and the fixed travel. Table 2 gives the variable travel for the same experiments. The variable travel approximately doubles when the density of the order doubles. The average variable travel becomes equal to the fixed part of the travel for a density of 75%.

Table 2. The Average Optimal Variable Traversal Travel

	DENSITY					
	5%	10%	20%	40%	80%	AVG.
WIDTH						
2	1.6	3.7	10.8	25.3	48.7	18.8
3	2.8	8.8	18.8	37.5	62.3	26.0
4	5.0	11.5	23.6	46.3	73.7	32.0
5	6.3	14.1	31.9	57.1	85.1	39.1
AVG.	3.9	9.5	21.2	41.8	67.5	28.8

The optimal split traversal policy consists of picking all items on one side of the aisle on the first traversal trip and all the items on the other side of the aisle on the second traversal trip. For long aisles, its length is approximately twice the fixed traversal travel.

4. OPTIMAL RETURN PICKING POLICY

With a return picking policy the picker enters and exits the aisle at the same end. The two types of return policies are illustrated in Figure 2. The split return policy occurs when in picking an order which contains items from more than one aisle, it is be advantageous to enter the aisle from one end, pick some of the items and leave at the same end it entered. Later in the tour the picker enters that aisle again from the other end, picks the remaining items and leaves again at the end it entered.

For the standard return policies, the optimal picking sequence is very simple to construct. The optimum picking sequence is to pick all items on one side, cross to the last item on the other side the pick all items on that side. This is illustrated in Figure 8. The optimality of this tour follows from the following well known property.

Property - Barachet (1957)

If all the points of a set lie on the boundary of the convex hull, or BCH, of this set of points, then this BCH is the shortest Euclidean distance Hamiltonian circuit of those points.

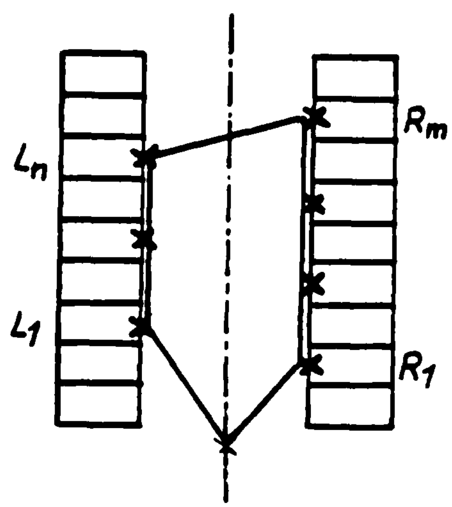


Figure 8. Optimal Standard Return Policy.

The optimal split return policy is more difficult to determine. The items must be divided into two sets. One set is picked from one end and the other set is picked from the other end. It is easily shown that an optimum split return policy from the near end would never include R_j or L_j without including R_{j-1} or L_{j-1}. Hence each subset can be optimally picked by an optimum standard return policy. Also all possible subsets can be determined by determining the item on the left and the item on the right where the tour ends. Complete enumeration of all possible combinations requires a computation effort which is quadratic in the number of items in the order. This is what was done in the experiments.

The set of experiments for the traversal policy was repeated for the return policies. In this case only the number of items was a significant factor in the travel distance at the 0.001 level of significance. The split return policy was, on the average, 5% better than the bottom or top return policies. The following tables give the average travel distance for the policies. The first set takes the average of the return policies from each end.

Table 3. The Average of Standard Return Policies from Each End of the Aisle

	DENSITY					
WIDTH	5%	10%	20%	40%	80%	AVG.
2	96.9	110.7	114.6	119.9	122.1	112.9
3	103.6	114.0	119.0	120.9	123.2	116.2
4	107.7	115.4	119.7	122.0	124.0	117.8
5	103.9	116.6	121.5	124.2	127.0	118.6
AVG.	103.0	113.5	118.3	121.8	124.1	116.1

Table 4. The Average Split Return Travel

	DENSITY					
	5%	10%	20%	40%	80%	AVG.
WIDTH						
2	97.2	101.0	109.0	115.3	119.6	108.4
3	94.6	100.0	106.4	114.7	121.5	107.5
4	92.7	100.6	109.6	119.1	125.2	109.4
5	96.8	107.7	115.4	121.8	128.6	114.1
AVG.	95.3	102.4	110.1	117.7	123.7	109.8

The best of the return policies was also compared on a case by case basis with the optimal traversal policy. The travel by the best of the return policies is given in Table 5. The best of return policies was, on the average, 23% longer than the optimal traversal policy. The breakeven point, when both policies performed equally well, was dependent both on the density of the order and on the width of the aisle. The density of the breakeven point decreased as the aisles became wider. The average density of the breakeven point was 75%.

Table 5. The Best of the Return Policies Travel.

	DENSITY					
	5%	10%	20%	40%	80%	AVG.
WIDTH						
2	73.4	97.8	107.5	114.8	119.6	102.6
3	88.6	99.5	105.9	114.7	121.2	106.0
4	85.2	97.3	109.6	118.6	123.4	106.8
5	83.6	104.0	113.3	121.1	126.6	109.7
AVG.	82.7	99.7	109.1	117.3	122.7	106.3

Table 6. Percentage Increase in Travel for Best of Return Versus Optimal Traversal Travel

	DENSITY					
	5%	10%	20%	40%	80%	AVG.
WIDTH						
2	17	51	50	33	9	32
3	39	43	33	17	- 2	26
4	29	34	30	11	- 8	19
5	24	39	22	2	-13	15
AVG.	28	42	34	16	- 4	23

To complete this analysis it is worthwhile to determine how much difference there is between the optimal traversal policy and a good heuristic traversal policy. One of the most widely used traversal policies is discussed in the next section and compared with optimum.

5. OPTIMAL FIXED SEQUENCE PICKING POLICY

A fixed sequence or fixed route picking policy is a policy which orders all the slots or locations on both sides of the aisle at one time. The items of any actual order are then picked following this fixed sequence. The main virtue of a fixed route is that the picking sequence for each individual order is very easy to determine since all slots are preordered. The disadvantage is that the required travel is not necessarily minimal.

In industry a much used fixed sequence is a repetitive Z pattern, as illustated in Figure 9. The pattern length X is an integer number of slots.

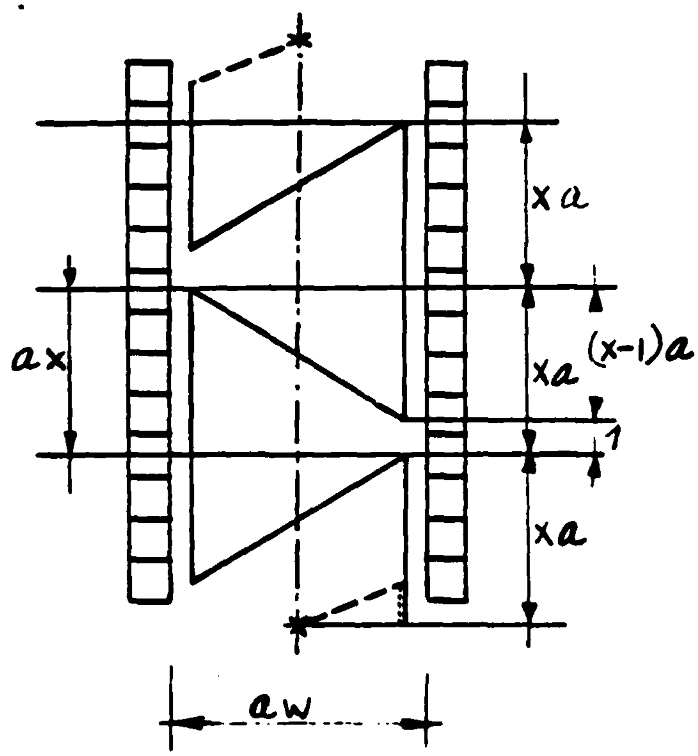


Figure 9. Repetitive Z-Pick Pattern

It is possible to compute the optimal pattern length assuming that all slots have to be visited. This is equivalent to the situation where all orders contain 2M different items or the density is 100%.

Let TH(X) be the total travel required by a pattern of length X, where X is an integer factor of M. Let TE be the travel from the entry point to the first item plus the travel from the last item to the exit point. TE is independent of the pattern length. TE is drawn in Figure 9 as a dashed line. Observe that TE compensates for the one unit length that the pattern travel is to long. This one unit compensation is indicated in Figure 9 by the dotted line.

$$TE = 2 \cdot SQRT(W^2/4+1)-1 \tag{9}$$

$$TH(X) = (M/X) \cdot (2X-1+SQRT(W^2+(X-1)^2)+TE \tag{10}$$

In order to find the optimum X with limited effort, the function TH(X) must satisfy certain conditions. In Appendix B it is proved that TH(X) is quasi-convex and differentiable, hence the optimal X can be found by setting the first derivative equal to zero.

In reality no order has a 100% density, but its items are still picked in the sequence that is optimal for a 100% dense order. The picking policy thus becomes heuristic for orders with a density less than 100%.

The same set of experiments run for the optimal traversal travel was also run for the heuristic Z-pick travel distance. These results are given in Table 7. The fixed sequence travel, is on the average, 12% longer than the optimal traversal travel. The difference is maximal for

a width of four and for an order density of 40%, at which time the relative difference is approximately equal to 30%.

Table 7. Increase in Travel for Z-pick Versus Optimum Traversal Travel

	DENSITY					
	5%	10%	20%	40%	80%	AVG.
WIDTH						
2	0.3	1.0	6.5	9.7	11.1	5.7
3	1.4	7.3	10.0	17.2	11.4	9.5
4	8.4	13.4	20.1	28.5	18.3	17.2
5	6.5	14.3	22.5	20.5	11.2	15.0
AVG.	4.2	9.0	14.8	18.3	13.0	11.9

6. CONCLUSIONS

The problem of determining the optimal picking sequence for a single aisle can be solved very efficiently both on large and micro computers. The computation times are in the order of a few seconds for orders up to a 100 items. This small computational effort for obtaining the optimal solution makes the use of heuristics unnecessary, if the data are available in machine readable form. The optimum fixed sequence Z-pick is very suitable for a completely manual system but results in a substantial increase in distance (up to 30%) over the optimum traversal policy.

For traversal policies both the density of the order and the width of the aisle are important to determine the travel distance, for return policies only the density is important for all practical widths. Except for very high densities the traversal policy yields a shorter travel distance than the return policies. The density breakpoint at which a return policy becomes more efficient, decreases with larger widths, from 75% for width 3 to 50% for width 5. Thus for most practical densities the traversal policy is better. If the order density is close to the breakpoint, it is best to compute the travel for both policies and to select the minimum option.

APPENDIX A

COMPUTER IMPLEMENTATION OF THE TRAVERSAL ALGORITHM

A specialized algorithm for finding the optimal traversal travels as the shortest path in a acyclic graph can be constructed, which implicitly incorporates the structure of the graph. Let $f(R_i,L_j,k)$ be the travel required to pick all the remaining items optimally and to exit the aisle, given the system is currently in state (R_i,L_j,k).

<u>Algorithm 1</u>

$f(R_m,L_n,L) = t(L_{n+1},L_n)$

$f(R_m,L_n,R) = t(R_{m+1},R_m)$

for i = m-1 to 1 with step -1

$f(R_i,L_n,L) = f(R_{i+1},L_n,R) + t(R_{i+1},L_n)$

$f(R_i,L_n,R) = f(R_{i+1},L_n,R) + t(R_{i+1},R_i)$

next i

$f(R_0,L_n,L) = f(R_1,L_n,R) + t(R_1,L_n)$

for j = n-1 to 1 with step -1

$f(R_m,L_j,L) = f(R_m,L_{j+1},L) + t(L_{j+1},L_j)$

$f(R_m,L_j,R) = f(R_m,L_{j+1},L) + t(L_{j+1},R_m)$

for i = m-1 to 1 with step -1

$f(R_i,L_j,R) = \min(f(R_{i+1},L_j,R) + t(R_{i+1},R_i),$
$f(R_i,L_{j+1},L) + t(L_{j+1},R_i))$

$f(R_i,L_j,L) = \min(f(R_{i+1},L_j,R) + t(R_{i+1},L_j),$
$f(R_i,L_{j+1},L) + t(L_{j+1},L_j))$

next i

$f(R_0,L_j,L) = \min(f(R_0,L_{j+1},L) + t(L_{j+1},L_j), f(R_1,L_j,R) + t(R_1,L_j))$

next j

$f(R_m,L_0,R) = f(R_m,L_1,L) + t(L_1,R_m)$

for i = m-1 to 1 with step -1

$f(R_i,L_0,R) = \min(f(R_i,L_1,L) + t(L_1,R_i),f(R_{i+1},L_0,R) + t(R_{i+1},R_i))$

next i

$f(R_0,L_0,L) = f(R_0,L_1,L) + t(L_1,L_0)$

$f(R_0,L_0,R) = f(R_1,L_0,R) + t(R_1,R_0)$

$f^* = \min(f(R_0,L_0,L),f(R_0,L_0,R))$

This specialized algorithm was programmed both on a CYBER 6400 mainframe and on an IBM PC in PASCAL to compare solution times. The sorting of the items was executed by the Quicksort procedure, Singleton (1969). The resulting computation times in seconds are given in Table 8 and are the average of three experiments. For each computer the first column gives the times for the computation phase (excluding input and output) and the second column gives the total run time. On both computers all debugging checks were turned off to generate the fastest running code.

Table 8. Computation Times on Different Computing Systems

	CYBER		IBM PC	
items	comp	total	comp	total
6	0.01	0.02	0.02	0.59
12	0.02	0.03	0.06	0.88
24	0.08	0.09	0.28	1.58
48	0.23	0.32	1.06	3.28
96	1.14	1.19	4.14	8.31

The IBM PC used a Intel 8087 numerical coprocessor to execute all real arithmetic. The IBM PC used a 64 bit real format and the CYBER a 60 bit real format.

The following observations can be made about the computation times. The ratio of the actual computation times is equal to 3.64, i.e. the IBM PC is less than four times slower than the mainframe. This is a very favorable ratio for the PC, considering the relative cost price of CPU time on each machine. On the PC the input and output operations required more CPU time than the actual computations. Finally the small running times on mainframe and PC make computing the optimal solution very feasible in a real life environment.

APPENDIX B

QUASI-CONVEXITY OF Z-PICK FUNCTION

Recall from expression (10) that

$$TH(X) = (M/X) \cdot (2X-1+SQRT(W^2+(X-1)^2)+TE$$

<u>Property 4</u>

TH(X) is a differentiable and quasi-convex function.

Proof. Let S_a be the level set of TH(X) with value a. Then

$$S_a = (X : X > 0, TH(X) < a) \qquad (11)$$

Substitution of (10) in (11) generates the following equation for the level set.

$$((a/M-2)^2-1)X^2 + 2(a/M-1)X - W^2 > 0 \qquad (12)$$

This is the equation of a parabola, which goes through the point with coordinates $(0,-W^2)$. Several cases arise depending on the sign of the coefficient $(a/M-2)^2-1$, which are illustated by Figure 12.

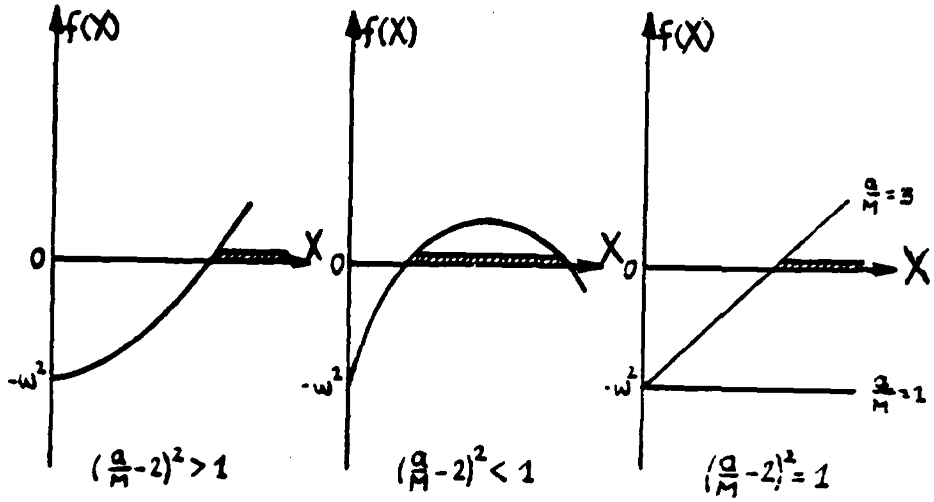


Figure 10. Level Set of Fixed Sequence Travel.

In each case the set of X values which generates positive function values is a convex set. Thus the level set is convex and TH(X) is a quasi-convex function. TH(X) is also differentiable for all $x > 0$. The global minimum is then found by setting the first derivative equal to zero. q.e.d.

This yields the following optimal pattern length.

$$X^* = (W^2+1)/2 \tag{13}$$

In the derivation of this formula, it was assumed that an integer number of patterns fit exactly in the length M of the aisle. The previously determined X^* might not satisfy that condition. The quasiconvexity of TH(X) makes it sufficient to search to the left and right of X^* for the first integer factor of M on each side, say X_L and X_R. The minimum of $TH(X_L)$ and $TH(X_R)$ is the optimal solution.

The optimal fixed sequence travel $TH(X^*)$ is equal to

$$TH(X^*) = M \cdot (3-2/(W^2+1))+TE \qquad (14)$$

This value is a lower bound on the optimal fixed sequence travel. The lower bound can be achieved when X^* is a factor of M.

REFERENCES

1. Barachet, L. L. (1957). "Graphic Solution of the Traveling Salesman Problem," Operations Research, Vol. 5, pp. 971-875.

2. Cutler, M. (1980). "Efficient Special Case Algorithms for the N-line Planar Traveling Salesman Problem," Networks, Vol. 10, pp. 183-195.

3. Garey M., R. Graham, and D. Johnson (1976). "Some NP-Complete Geometric Problems," Proceedings 8th SIGACT Symposium on the Theory of Computing, pp. 10-22.

4. Gilmore, P. and R. Gomory (1964). "Sequencing a One Step-Variable Machine: A Solvable Case of the Traveling Salesman Problem," Operations Research, Vol. 12, pp. 655-679.

5. Karp, R. (1972). "Reducibility among Combinatorial Problems," Complexity of Computer Computations, pp. 85-104, Miller R. and Thatcher T. (eds), Plenum Press, New York.

6. Lawler, E. (1971). "A Solvable Case of the Traveling Salesman Problem," Mathematical Programming, Vol. 1, pp. 267-269.

7. Ratliff, H. and A. Rosenthal (1983). "Order Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem," Operations Research, Vol. 31, No. 3, pp. 507-521.